DOCUMENT RESUME

ED 129 299 IR 004 084

AUTHOR Lower, Stephen K.
TITLE Making C.A.I. Make a Difference in College Teaching.
PUB DATE Jul 76
NOTE 35p.; Paper presented at the North Atlantic Treaty Organization Advanced Study Institute on Computers in Science Education (Louvain-la-neuve, Belgium, July 1976)

EDRS PRICE MF-$0.83 HC-$2.06 Plus Postage.
DESCRIPTORS Chemistry Instruction; *College Instruction; *Computer Assisted Instruction; *Failure Factors; *Instructional Innovation; Programing Languages; Tutorial Programs
IDENTIFIERS Simon Fraser University

ABSTRACT

An explanation for the failure of technology and computer-assisted instruction (CAI) in particular to make much headway in education is that even when innovations are introduced in the classroom, their potentials are not exploited; rather, they are used in traditional ways. The integration of new technologies with other classroom activities is also necessary to realize any advantages. This has not generally been done. Experience with CAI in the chemistry department at Simon Fraser University in British Columbia began in 1969. An interactive program was developed to review basic problem sets with first year students. This program was used in conjunction with explanatory audiotapes. A second drill and practice series was also developed. The materials made little difference in student performance until the traditional structure of the course was changed. The chemistry department gained significant experience with hardware and course development. They exchanged IBM's Coursewriter III system for York/APL which in turn they abandoned for IBM's APLSV which still was inadequate for authoring and delivering CAI. Eventually the department developed its own authoring language which reflected the needs perceived in six years of CAI experience. A 17 item bibliography is appended to the report. (KB)

****
* Documents acquired by ERIC include many informal unpublished *
* materials not available from other sources. ERIC makes every effort *
* to obtain the best copy available. Nevertheless, items of marginal *
* reproducibility are often encountered and this affects the quality *
* of the microfiche and hardcopy reproductions ERIC makes available *
* via the ERIC Document Reproduction Service (EDRS). EDRS is not *
* responsible for the quality of the original document. Reproductions *
* supplied by EDRS are the best that can be made from the original. *
****

U S DEPARTMENT OF HEALTH, EDUCATION & WELFARE
NATIONAL INSTITUTE OF EDUCATION

THIS DOCUMENT HAS BEEN REPRODUCED EXACTLY AS RECEIVED FROM THE PERSON OR ORGANIZATION ORIGINATING IT. POINTS OF VIEW OR OPINIONS STATED DO NOT NECESSARILY REPRESENT OFFICIAL NATIONAL INSTITUTE OF EDUCATION POSITION OR POLICY

MAKING C.A.I. MAKE A DIFFERENCE IN COLLEGE TEACHING

Stephen K. Lower

Dept of Chemistry, Simon Fraser University
Burnaby, B.C. V5A 1S6, Canada

ABSTRACT. The general theme of this presentation is that if tutorial-type computer-assisted instruction (CAI) is to become an accepted and viable means of learning, then its application to that task must result in a significant improvement in the cost-effectiveness of the overall process. Even the best possible CAI program cannot accomplish this unless the immediate outcomes, such as improved achievement or shorter learning time, can themselves be exploited in a useful and meaningful way.

I shall begin by commenting on instructional innovation in general, and on the failure of innovative technology to make much of an impact on teaching. Successful introduction of a new instructional medium usually requires adjustments in the use of other methods, including the classroom experience itself. Thus the manner in which CAI is used seems to be at least as important in determining the "success" of CAI as is the quality of the CAI program itself. CAI should always be regarded as only one component of the course, and must be sensitively interfaced to the other components, such as audiotapes, reading, and lectures.

Some of these points will be illustrated in Part II of the presentation, which describes the use of audio-tutorial and CAI methods that have been used in the teaching of Chemistry at SFU. Details are given of an introductory course in Chemistry in which student performance was improved, not only in that course, but also in a following, "traditionally taught" course as well.

IR 004 084

This is the text of an invited presentation delivered to the NATO Advanced Study Institute on Computers in Science Education, Louvain-la-neuve Belgium, July 1976. *Copyright 1976 by Stephen K. Lower.*

"PERMISSION TO REPRODUCE THIS COPYRIGHTED MATERIAL HAS BEEN GRANTED BY

Stephen K. Lower

TO ERIC AND ORGANIZATIONS OPERATING UNDER AGREEMENTS WITH THE NATIONAL INSTITUTE OF EDUCATION FURTHER REPRODUCTION OUTSIDE THE ERIC SYSTEM REQUIRES PERMISSION OF THE COPYRIGHT OWNER."

I tend to view CAI as a communications medium, and thus as something that is more the art of the teacher than of the programmer. Both skills are necessary; it seems that the best CAI materials, like the best textbooks, will be written by a fairly small number of authors who have the facilities and especially the time required to design and develop instructional programs.

Part III will take up the matter of authors, author languages and program transfer. One problem to be given particular emphasis is my feeling that CAI is not evolving as rapidly as computer technology and computing practice in general. Although a few systems are exploiting the current state of the art, the majority still employ primitive and inadequate hardware-- in all too many cases, leading to a poor image of CAI. The most serious thing standing in the way of CAI at the present time is the dearth of effective and well-documented instructional programs. This is partly an effect of the deficiencies of the current major authoring languages, which are unable to serve as practical vehicles for the design and presentation of CAI that is sufficiently sophisticated to be cost-effective.

PART I. EDUCATION AND INNOVATION

In many ways, the classroom has always been something of a backwater of technological change. Overhead projectors, which were frequently used for displaying scores in bowling alleys in the 1940's, did not become widely available in most schools until twenty years later-- and even today, most university teachers seem to be rather wary of them, still apparently preferring the familiar, venerable, and dusty chalkboard, in spite of its limited visibility to the majority of the occupants of large lecture theatres. Ten years ago portable cassette recorders became inexpensively available on a sufficiently large scale that most teenagers could acquire them. But audiotape-based learning material, in spite of its demonstrated efficacy, plays only a minor role in most classrooms, even while the far less effective and more costly "school broadcasts" continue to emanate from the transmitters of the Canadian Broadcasting Corporation and similar national authorities in other countries.

IS 00-1084

The situation is even more striking when one considers the two technologies that have most radically altered the nature and scope of communicatons in our society during the past twenty years: television and the computer. Although closed-circuit television has been available to many schools since the late 1950's, this medium has made very little real impact on the way things are done in the classroom, even while in the real world outside the school, television has managed to transform many aspects of the education of our children, and in some respects has left the classroom far behind, having succeeded in imparting to many children a view of the world whose scope and sophistication extends far beyond that of the school curricula.

The case of the computer is somewhat different. It has invaded most schools, but only as far as the business office; to most school administrators, "computer" means EDP-- electronic data processing. Occasionally the computer manages to become a subject of the curriculum itself, but even then it is presented more as a tool for business data processing or applied mathematics. It is true that students in universities are now beginning to be exposed to the use of the computer as a tool for problem solving and simulation. Although much effort has gone into developing and distributing programs for such purposes, there is very little published evidence showing any significantly increased learning by this means. In contrast, there is considerable (albeit not always unequivocal) evidence for the efficacy of tutorial type CAI, but this represents a minor portion of instructional computing applications at most universities. A possible reason is that the problems associated with the design and implementation of this style of CAI do not tend to engage the interests of those who are already involved in computer use in a given discipline. The result is that as an instructional device in its own right, the computer is generally regarded as a highly experimental, expensive, and rather doubtful luxury that had best be postponed for the indefinite future.

There is no single, simple reason for the failure of these technologies to make much headway in education; numerous factors, such as the sometimes ambiguous role of educational institutions in our society, the way teaching and learning are considered to enter into instruction, the nature and sociology of the teaching profession-- these and other considerations all enter into the problem in a complex way. All we do know is that somehow, there has been very little real incentive to introduce innovative technology and to make it work in the school environment. In contrast, the non-school instructional environments of industrial and military training have generally been more receptive to innovative and non-traditional teaching methods.

One factor that does seem to emerge, however, is that until recently there has been relatively little external pressure on our schools to make any significant improvements in the results or efficiency of instruction. "Cost effectiveness" has always been a bad word in educational circles, and has only limited meaning in much of higher education anyway, owing to the relatively small and advantaged cross-section of the population that colleges and universities have served; "good" students, by and large, are generally quite capable of instructing themselves; all they require is stimulation and direction, and post-secondary institutions have usually been able to meet the needs of such students without much attention to instructional methodology. For the best of these students, considerations such as student-faculty ratio, reputation of the faculty, and the size of the Library have been and are valid indicators of the overall "quality" of the institution.

These and other traditional criteria began to lose their universal validity during the far-reaching social changes that commenced in the 1960's, when the responsibilities of our institutions of higher learning were extended to a much broader element of our population-- and one that was in many ways ill-prepared academically. At a time when better preparation of students was required at all levels, severe economic constraints have led to a deterioration in performance standards, only partly disguised by grade inflation in many jurisdictions. Partly in response to social and political pressures (including the need to maintain sufficient students to attract the funding they require), many of our colleges and universities have reduced their entrance standards, either by eliminating entrance examinations, or accepting secondary-school grades that have been steadily declining in meaning.

It is only recently that this breakdown of our public education system has begun to attract political attention, elicited in part by newspaper accounts of how the majority of first-year students at several major North American universities were unable to compose English prose at an acceptable level. (That skills in basic arithmetic and mathematics may have declined even more has not as yet been as widely reported.) In the State of New York, the public and political reaction has resulted in the passage of legislation requiring secondary school (Grade 12) graduates to be able to read and write at the Grade 9 level. It seems inevitable that similar pressures for performance improvement will be brought to bear in other jurisdictions, and that our educational institutions will be forced to address themselves to instructional efficiency in a direct way, as the public comes to doubt that simply hiring more teachers, even if this were fiscally possible, is unlikely to lead to any significant alleviation of the problem.

Although the development of external pressure for better teaching may provide the primary driving force for the incorporation of improved technology into our schools, it would be quite wrong to believe that the lack of such pressure has actively impeded technological change in the past. Given even the most progressive attitude and the best of intentions, there are still two major potential energy barriers that must be surmounted before the new technology, once introduced, can be expected to take hold and grow. One of these is simply learning to make effective use of the new methodology. The other, which I will discuss first, concerns the problem of exploiting the benefits of the new method in a way that is meaningful to both the student and the school.

WILL THE DIFFERENCE MAKE A DIFFERENCE?

A number of new instructional techniques-- most notably CAI-- have been introduced and used under conditions where improved learning, shorter learning times, and reduced dropout rates have been clearly documented. Why, then, have these methods not been more widely adopted-- are these benefits not consistent with the goals of our educational establishment?

In some cases (particularly our more prestigious universities), perhaps not. I would like to offer one explanation, however, that my own experience suggests must have something to do with the failure of new methodology to take hold. It does not appear sufficient that a new instructional method lead to an improved outcome: this improved outcome must itself make a significant difference to the educational institution in terms of its own goals. Thus if the raison d'etre of a school, university, or academic department is to provide positions for scholars, or to simply "offer courses", then there will be little reason to go to the trouble of trying something new.

It comes as no surprise to see that, in the U.S., the most active areas of CAI have been in "remedial" instruction among disadvantaged children, and in medical and allied health fields training, where clearly defined educational goals and performance standards exist, and failure to meet them could have disastrous social and political consequences. Most of our academic departments in universities are interested primarily in the "best" students-- the ones who require the least amount of "teaching". Improving the effectiveness of instruction will have little effect on the number of "A" and "B" grades earned;

the principal benefit will be to the underprepared and potential failing students-- who are in any event likely to confer little if any glory on the department that produced them.

One can look at the time factor in improved instructional efficiency in a similar way. What of the fact that the time required for students to learn a given body of material can be substantially reduced? Do we allow this difference to make any difference to the student's overall progress? Our educational process is divided into quanta of grades and courses. Grade 2 is for learning just so much about writing, reading, and mathematics; beyond this the students coast along and try to keep themselves interested while the teacher concentrates on the slower half of the class. In the University, we go by semesters. Chemistry 101 is thirteen weeks long; if, by using a more efficient approach, I complete the material in nine weeks, what difference does it make? The students still have to wait for Chemistry 102 to begin before they can learn more chemistry.

Perhaps the only thing that will eventually decide the issue is the cost factor. It is clear that anything as vague as "cost of instruction" can be computed in innumerable ways, even in traditional teaching. Combining this with the hocus-pocus of computer costing creates enough variables to support anyone's point of view. There has, in the past, been a tendency to cite additive cost figures only-- that is, to consider traditional instruction as "free", so that any improvements such as CAI are add-on "extras". As long as the costs of computers and computing continue their decline at an annual rate in excess of 15%, it appears to be only a matter of time before the topics of educational seminars change from "cost effectiveness of CAI" to "cost effectiveness of teachers".

EFFECTIVE USE

The other major impediment to the adoption of innovative technology has to do with our learning to use it in an effective manner. Just as there is a natural tendency for teachers to teach in much the same way that they themselves were taught, teachers who are presented with a new instructional medium will frequently attempt to use it in a manner that is fundamentally the same as whatever other medium it replaced, without really trying to go beyond the limitations of the old medium and fully exploiting the added capabilities of the new medium.

This phenomenon can readily be observed in institutions which, like my own, have lecture halls equipped with overhead projectors. In the first place, many of the faculty, particularly the more senior ones, will not use the projector at all, even where its use presents a clear advantage to the majority of students in the room. Of those who do use the projector during their lectures, the majority will use it very much like they used the chalkboard: they transcribe their message from lecture notes to projector surface, and then the students are expected to carry out a second transcription from projection screen to notebooks, in the time honored tradition of the schola. Only a small number of faculty of my acquaintance take advantage of the fact that their notes can easily be copied onto transparent sheets before the lecture, so that they need not spend the time writing during the class period. Even fewer eliminate the second stage of a highly inefficient and imperfect process by supplying printed copies of the lecture notes directly to the students. I must confess that it took me several years to adopt these practices myself, but in my graduate photochemistry course, where a surprisingly large amount of time was taken just in writing subscripts and superscripts, this more effective use of the overhead projector enabled me to cover approximately 50% more material during the semester.

Many new (and even some old) users of computer-assisted instruction have similarly transferred the styles, limitations, and inappropriate aspects of other forms of instruction to this new medium. A common example is the use of the computer terminal as a device for presenting reading matter, when books or printed sheets would be far more appropriate, particularly when slow printing terminals are used. Another common failing is to transfer the contents of a programmed instruction text directly to CAI, with the effect that a more expensive medium is being used to present material in a fashion that is limited by the constraints of the programmed book.

In the sciences, one frequently encounters CAI lessons that consist of little other than a series of problems very much like the kind that are found in textbooks; the sole function of the program is to present the problem, and check to see if the student's answer is correct. All too often there is no attempt made to go beyond the space limitations of the text by providing a thoughtful, graded development of these problems through several stages of complexity, with special effort made to identify and analyze common sources of student error or misunderstanding.

An example that I have seen at a number of installations consists of a program that randomly selects the components of a chemical reaction, and then calculates the coefficients for the balanced equation. The student is supposed to enter the same coefficients; if he does not, he is simply told he is wrong-- no attempt is made to identify the error or the faulty reasoning that led to it, or to teach the proper approach to equation balancing. This kind of program is computationally elegant, but of questionable pedagogical value.

It is undoubtedly the flexibility of CAI that sets it apart from other instructional media; unless this capability is adequately exploited, it is unlikely that CAI will be worth the time, trouble, or cost. The degree to which this potential can be realized in a given situation will depend very strongly on the skill and sensitivity of the author, and on the power and sophistication of the authoring language-- a topic to which I shall return later in this article.

At the most elementary level, effective utilization of CAI's flexibility involves such things as providing for random selection not only of numerical parameters in problems, but also of the form and wording of the problem itself, so that students are not trained simply to solve certain set kinds of problems.

There should also be alternative means of presenting and accessing the same material, so that the same basic programs can provide instruction in various modes such as tutorial, drill, self-test, etc.

When these aspects of flexibility are combined with internal record keeping capabilities, one is then in a position to move to adaptive CAI, in which the rate and direction of each student's movement through the material is related to that student's preference and performance.

In order to gain the maximum possible benefit from CAI, a rather complete rethinking of the "structure" of the subject matter being taught is required. This is so because the structures we associate with many subjects are, at least in part, artifacts of the manner in which they are commonly taught. Because CAI can introduce so many new dimensions to learning, it is important that we take a careful look at our subjects from these different viewpoints. Ultimately, one would hope that it will become possible to emphasize different kinds of structural development that will correspond more closely to the needs of individual students. As one moves to more sophisticated terminal devices

that are able to produce graphics and film displays, animated motion, and audio responses, the possible ways of structuring the subject increase enormously, and present the possibility of developing alternative course structures attuned to different cognitive styles.

INTERFACING CAI TO THE COURSE

In this section, I want to comment on the second important source of difficulty in making CAI work-- the problem of integrating CAI into the rest of the course. My own experience is that the ultimate "effectiveness" of CAI seems to depend as much on the degree to which this integration is accomplished, as on the basic quality of the CAI material itself. Thus even the best-conceived CAI lessons, presented on suitable terminals, may have only a limited effect on the way in which the course is taught and on the ultimate performance of the students, unless it is carefully and sensitively interfaced to the course it is intended to support.

The most common mistake is to treat CAI merely as an instructional "aid"-- to simply add it on to an existing and unchanged regime of lectures, lessons, and assignments, without making any changes in the style or function of these other activities. CAI is a sufficiently powerful tool that it is bound to exert a perturbing effect on other instructional activities; if these effects are ignored, then CAI is likely to be perceived by the student as something of secondary importance-- as "extra work" for those who want it.

CAI is best viewed as one component of a course that consists of numerous other components such as lectures, group discussions, tutorials, reading and problem assignments, laboratory experience, audiotape-based media, etc. Each component should be used primarily for what it can do best, and all should be integrated in a comprehensive way. This implies more than that the course be "organized"; this organization must be clearly communicated to the students.

The common failure to accommodate other aspects of teaching to CAI seems to be widespread even at institutions that have an adequate CAI system, and seems to be a common reason for the failure of CAI to catch on outside the classes of a very few highly motivated instructors. One problem, of course, is that CAI is basically an "individual study", self-paced method of learning, and that many departments are simply not "set up" (organizationally or psychologically) for that kind of departure

from tradition. Often CAI is their first real experience in this area, and all to frequently they never really accommodate to the idea of self pacing. This is particularly likely to occur where the proponents of CAI are people whose primary interests are with computers rather than teaching.

The specific role that is assigned to the CAI component of a course will vary with the nature of the course, and also with the sophistication of the CAI material and of the terminal on which it is presented. Although CAI dialogs can be divided into many types, there are two that deserve particular attention here because they are so ideally suited to CAI, and their goals are usually less adequately realized in conventional class settings.

One of these is "drill and practice". Despite the unpleasant associations that this term may elicit in many memories, drill and practice represents one of the most important and difficult aspects of teaching. Its importance lies in the fact that most bodies of knowledge require the development of certain basic skills or associations that can only be built up through repeated exercise or pattern practice.

Drill and practice is difficult to present by conventional classroom means. Like any kind of exercise, it is most efficient when both its difficulty and rate are individually tailored to the needs and progress of each student. In the classroom, the teacher must aim for some intermediate group of people, with the result that both the better and the poorer students become bored and discouraged. In addition, drill and practice does not represent a very inspiring or creative application of most teachers' skills, either in the presentation or in marking the results of written drill. The difficulties associated with drill and practice become increasingly serious as classes become larger and the diversity in student background and ability widens. It is very likely that the serious deterioration in basic math and English skills among high school graduates that has recently come to public attention in North America is due, in large measure, to the impracticality of including effective drill and practice in conventional classroom environments.

It must be clearly understood, however, that merely transferring this mode of learning to CAI will not provide any automatic remedy to the situation. Here again, there is a need to completely reconsider the whole process, and to make sure that the computer is furnishing instruction that is significantly more adaptive and interesting than before. It is perhaps unfortunate that drill and practice can be programmed quite easily on the simplest computer systems, which are not always

capable of offering very much improvement over the more traditional methods.

The other dimension of CAI that goes beyond other forms of instruction is simulation. Of course there need not be any fundamental distinction between drill-and-practice and simulation; that is, drill can be based on simulation. Generally, however, simulation involves not so much the development and exercise of specific basic skills as it does the sharpening of the student's ability to select and apply these skills in various situations. Essentially, simulation should be designed to broaden a student's experience-- to provide a "feeling" for the application of principles in a more general context. As such, it represents an activity that is often a closer approximation to the final desired outcome of the instructional process.

As with other forms of CAI, however, it is quite possible to write an elegant simulation program that has little real pedagogical value, owing to inadequate integration of the simulation exercise with the rest of the course. With batch programs, particularly, one must always ask whether the same concepts might not be better conveyed through a few carefully designed and well printed sheets of paper. The contrast is particularly apparent when one attempts to use a slow, noisy printing terminal such as a Teletype to construct a series of graphs.

It is often found that the computational parts of the program are far easier to write than is the documentation required to make them very useful, and too few authors take the time to do this. The best simulations are, of course, highly interactive, and are probably best presented within the framework of a conventional CAI system where these interactive features can be readily utilized.

Part II. CAI AND CHEMISTRY AT SIMON FRASER UNIVERSITY

The purpose of this section is to summarize the experience we have had in developing and applying several innovative instructional methods, particularly CAI, at my own institution. As will be seen, this history depicts a "real", as opposed to an ideal situation that departs quite markedly from what one would ordinarily recommend, and certainly from what might have happened if we had possessed prior experience.

Simon Fraser University is a relatively new institution, having been founded in 1965 as the third public university in British Columbia. Unlike our much larger sister institution, the University of British Columbia, we have no professional schools such as Engineering, Agriculture and Medicine that traditionally supply the greater part of first-year students in the Sciences. As a consequence, the enrollments in such courses tend to be small-- less than 200 in the Fall, and around 20-50 in the low-enrollment Summer trimester. Although we are not faced with the pressure of large numbers of science students, we do suffer from the necessity of accommodating a wide variety of student backgrounds and interests in a relatively small number of courses, since providing different course "streams" for different groups of students would not be economical.

At the time SFU opened, considerable encouragement was given faculty to explore new and innovative teaching methods, and although that atmosphere unfortunately failed to endure beyond the first few years of our existence, it was the essential first step that attracted my own participation in the development of instructional methodology.

Our first major effort was the use of audiotapes as supplements to both laboratory and lecture courses. An attempt [1] was made to apply Postlethwaite's technique [2] to the chemistry laboratory course, but our initial efforts met with only limited success and this aspect of audiotape use was dropped after a few years. The more successful application of audio-tutorial methods occurred in conjunction with the theory part of the course. It bears commenting on here because it established the environment that CAI eventually came to occupy; in a sense, there is a direct evolutionary path between the audiotape work and our first work with CAI several years later.

My principal use of audiotapes commenced soon after my first semester of teaching, when I realized that an undue amount of time was being spent, both inside and outside of class, dealing with the same basic set of routine questions relating the the weekly problem sets that are commonly assigned in First-Year Chemistry courses.

I therefore began making tape-recorded commentaries on each problem, explaining what the problem is about, what chemical principles are involved, and outlining a general approach to the solution-- but certainly not simply dictating the solution to the student, which would clearly have little real learning value. These tapes, together with printed notes that constitute an integral part of the method (and which require students to fill in blanks as they go along, thus introducing an active

element into the activity), were made available in the Library for use on a purely voluntary basis, and received widespread acceptance by the students, as evidenced both by their responses to course-evaluation questionnaires, and their degree of use of the tapes.

Not only was my original goal achieved (fewer students asking questions of the type "How do you do Problem 4?"), but there is good evidence [3] that the availability of these tapes led to improved performance by the weaker students in the course examinations. Many students come to our introductory course with very little ability to analyze and solve simple numerical problems. It is not so much a matter of "mathematics" as it is "problem-solving ability" that is lacking; it may be that the tapes help these students to get the practice that is essential for developing confidence and proficiency in problem solving.

Although it was the weaker fraction of the class that obviously benefited most from the tapes in terms of grade improvement, it was interesting to note that the abler students nevertheless made very heavy use of the tapes, but more for review purposes than because they were "stuck". This phenomenon has been consistently observed in all of our work in instructional innovation, including CAI.

These tapes have since been extended and generalized to cover eighteen sets of problems common to most First Year courses [4]. This has extended their utility to students in other courses, not only at SFU, but also in over forty high schools and colleges in British Columbia and elsewhere.

Another major audiotape project currently underway is a series of units covering various aspects of chemical bonding and structure. In these I am attempting to exploit the combination of the spoken word and the printed page to accomplish what neither medium can do alone. An audio commentary can draw the student's attention to particular features of an equation or a diagram in a far more compelling and immediate way than can text printed on an adjoining page or in a caption. By the judicious use of halftone screening and two-color printing, the spoken commentary can effectively "dissect" a diagram containing a great deal of information.

As was pointed out in the preceding section, the use of a new kind of medium usually calls for the development of new techniques, rather thanthe mere transfer of older methods. In the case of audiotapes, it may require some time and effort (and perhaps some expert coaching) to develop a proper speaking style. Audiotapes are not lectures, and they should not sound

like lectures-- and they should certainly not sound like someone simply reading a text to the student. Similarly, the design and layout of the printed notes and the construction of diagrams must be carefully adapted to the presence and nature of the spoken commentary; styles that might suffice quite well for a conventional textbook may be quite inappropriate in this kind of application. It might be added that I have not found that audiotape material can be developed in significantly less time than CAI lessons; I generally estimate that 50 hours of work is required to produce a 40-minute tape-based lesson.

BEGINNINGS OF CAI

Our experience with CAI began in 1969 when IBM Canada provided us with support for the installation of several terminals in the Chemistry Department. IBM's Coursewriter III CAI system was implemented on our 360/50 computer, and a small group of interested faculty began to write lesson materials and to explore the possible applications of CAI. Although some of the short courses that were written then did receive considerable use at first, this use soon dropped off as the novelty of the method (as seen by both students and faculty) abated. After about six months the terminals were mostly deserted, and served mainly as showpieces for visitors. In retrospect, the direct reason for the failure of our early efforts is fairly clear: we didn't really know what we wanted to do with CAI.

We then tried a somewhat different approach. Having felt that the problem-solving tapes served a purpose that could perhaps be even better served by an interactive method like CAI, I began constructing a series of CAI lessons that duplicated several sets of the CHEMICAL CALCULATIONS materials. The initial lessons were quite well received by the students, who used them quite heavily from the very start.

These problem-tutorial lessons grew to cover some sixty different problems, the whole collection becoming known as the CHEMEX CAI program. During the time that we continued to use Coursewriter, CHEMEX was by far the most popular and heavily used of our Chemistry CAI programs, and its popularity seems to have continued at the other institutions that have implemented our CAI materials.

The reason for the popularity of CHEMEX in comparison to many of our other CAI programs is not hard to understand. A student comes to this program for help on a particular problem-- one that he has attempted, but has not really understood. His

interest and readiness to understand are at a peak, and there is a reasonable chance that this motivation will yield a concrete result. In short, the students apparently feel that CHEMEX meets a real and immediate need.

It is interesting to note that many students apparently find the CHEMICAL CALCULATIONS audiotapes and CHEMEX complementary, in that the tapes, although they lack the interactive feature of CAI, are perhaps better able to generalize beyond the details of an individual problem and thus bring the relevant chemical principles into sharper focus. Thus even though the popularity of the audiotapes declined on a relative basis following the advent of CAI, there is reason to belive that both media fulfill useful and somewhat complementary needs, and I have continued to develop and extend both forms of the same basic material.

Our second major series of Chemistry programs consisted of about twelve programs containing mostly drill-and-practice (with a small tutorial component) covering several basic introductory topics. Most of the drill material made use of randomly-generated problems, something that is exceedingly difficult and awkward to do in Coursewriter. This collection, known as ALCHEM, and a remarkably elaborate program on organic hydrocarbon nomenclature (ORNOM), written by an undergraduate student, comprise (with CHEMEX) the major Coursewriter programs from SFU [5]. They have been implemented at over forty other installations, mostly in the U.S.; it is likely that this represents the broadest distribution of CAI programs in Chemistry that has yet been accomplished-- including programs whose development has been funded by large sums of money through the U.S. National Science Foundation. Since the Canadian Government does not provide funding support for educational development, we had to manage with the minimal funding we obtained from the University, which provided for little more than a small group consisting of myself and a part-time programmer assistant.

USING CAI IN AN OBJECTIVES-BASED COURSE

Although we managed to write a considerable amount of CAI material in the period 1969-1972, it was not really making any apparent difference to the overall performance of the students in the First-Year course. This was due partly to the small amount of material; we had not yet achieved anything near "critical mass", where CAI could be seen as an integral part of all twelve weeks of the course. It began to be clear, however,

that the more or less "traditional" structure of the course might be the limiting factor. In 1972, therefore, I began to make a basic change in my approach to teaching the introductory course. Briefly, I moved toward, but did not totally adopt, the so-called Keller approach. I switched from a "lecture based" system to one that is built around a set of well-defined performance objectives. The course was divided into nine sections, each ending with an examination. Each examination resulted in a letter grade, in which "C" required "50% or better" attainment of 75% of the objectives tested; students who achieved less than this were required to make up their failures by writing another exam.

This system differs from the strict Keller plan in several respects. First, the course is not really individually-paced; the examination schedule is set at the beginning of the semester, and failure to write an exam on the scheduledate results in a failing grade. This policy stems from my doubt that a true self-paced course can compete effectively for the student's time in an environment where all the other courses hold him to a rigid schedule.

A second departure from the Keller plan is that I did not do away with classroom lectures. I feel very strongly that the classroom "experience" is an important facet of university-level education-- as opposed to training, in which the Keller plan undoubtedly excels. On the other hand, I do not believe in giving any more lectures than are required to establish a context for the course material, to inject the cultural and humanistic aspects of the subject, and to hopefully provide some motivation. It is these, of course, that constitute the legitimate purpose of a lecture; by restricting the lectures mostly to these functions, I was able to get by with only about twelve lectures during the semester, instead of 30-36.

The third major difference is that I am still sufficiently old fashioned to believe student grades should, to some extent, reflect ability to achieve criterion performance on schedule. Consequently, students who fail an examination can earn no more than a "C" grade on re-writing it. Also, "A" grades are not awarded on midterm examinations. A student desiring this grade must write the final examination, which is marked in the usual "relative" manner. I view the "A" student as one who can synthesize ideas beyond the confines of specific learning objectives, and the final examination reflects this criterion.

Since I do not teach this course every semester, it is possible to draw some conclusions about the effectiveness of this approach by comparing various aspects of student performance

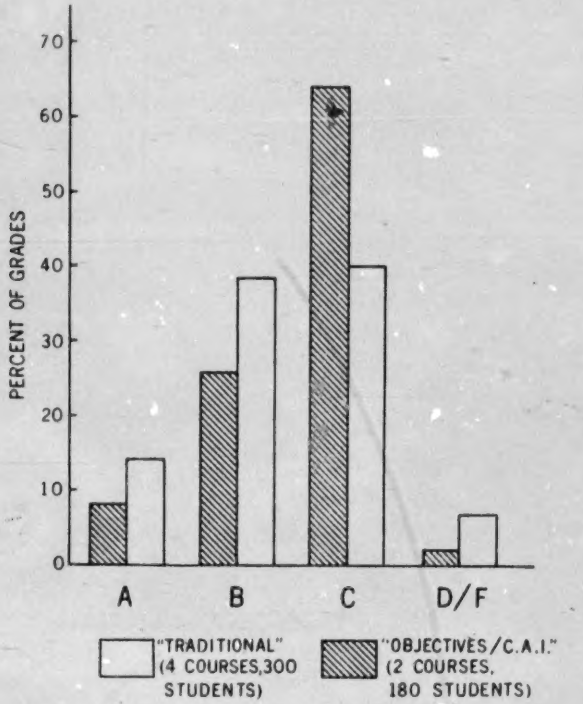

FIGURE 1. Course grade distributions for the Introductory Chemistry course. Several different instructors were in charge of this course during semesters when it was "traditionally" taught.

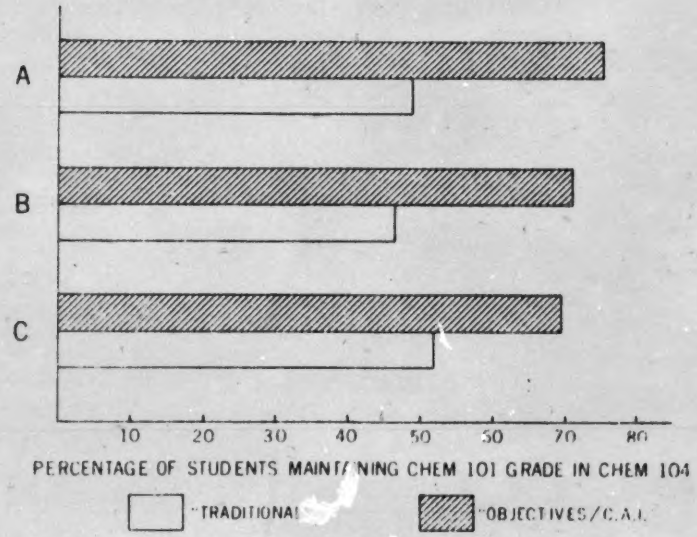

FIGURE 2. Fraction of students receiving a given grade in the Introductory Course, who were able to maintain or improve that grade in the subsequent, more rigorous course.

between semesters in which the course is taught by "my" method and by conventional methods. Making the comparison only between semesters in which enrollments exceeded 80 students, we obtained the grade distributions shown in Figure 1. The very small number of "failures" (D and F grades) is largely a consequence of my compelling students to pass examinations in each section of the course. The distribution of B and C grades is probably an artifact of my "absolute" grading system. The number of A grades, while perhaps lower than that in the other courses, is about the same as I have always given, even when I taught the course "conventionally", and may simply represent my own personal view of what an "A" student is.

Since we do not have standardized examinations or grading practices in our Department, it is difficult to assess the "validity" of the grades from semester to semester. One way of getting at this might be to ask, "how many of your students can maintain or improve the grade you gave them, in the next course they take in the same subject?" This approach seems especially appropriate in this introductory course, whose main purpose is to prepare students for our standard one-year Chemistry sequence.

Figure 2 shows the comparative percentages of students in the two kinds of courses that were able to maintain or improve their Chemistry 101 grade in the subsequent course. It is seen that ordinarily, less than half of the students leaving Chemistry 101 with a passing grade are able to maintain or improve that grade in Chemistry 104. By contrast, approximately 70% of the students in the "objectives-based" course were able to achieve this criterion.

What do these results say about the "effectiveness" of CAI? By themselves, I believe they say very little. The most important and significant aspect of the course is its objectives-based approach. Since many of these objectives relate to particular kinds of skills, the role of CAI is to make it easier for students to develop these skills with a minimum expenditure of time. In other words, the use of CAI is making it possible to present the course in a more effective manner; the "effectiveness" of the CAI itself, while not unimportant, is only secondary.

One point worth noting is that the CAI materials I have developed do not seem to meet the needs of a small group of under-prepared students. These people still must come to me for assistance. However, since CAI has largely relieved me of the necessity of personally ministering to the needs of most of the "average" students, CAI is benefiting these weaker students indirectly by giving me the time to provide them with the more

personalized help they require. This may be another important reason for the small number of failures. The fact that these students are receiving more of my attention also helps me gain a better insight into their learning difficulties, and thus may help be improve the design of the learning materials.

HARDWARE AND SOFTWARE

Our first CAI work was based on IBM's Coursewriter III system, implemented initially on a 360/50 and later on a 370/155 computer. The terminals were 2741 typewriter terminals which provided both upper and lower case printing at around 14 characters per second. We found that six of these terminals could support a class of approximately 100 students at our level of use. It seemed to us that the difficulties in getting students who are not familiar with typewriters to enter complicated equations would be nearly as great as teaching them the theory itself. Consequently we avoided many areas of Chemistry such as equilibrium and bonding that would have required the input of complex expressions or the presentation of structural diagrams. We also avoided using the relatively slow and noisy terminal to print graphs or large amounts of text. While we would have preferred CRT terminals to printers, we did observe that hard copy has some advantages, and we tried to design our CAI materials to allow students to exploit these advantages. The CHEMEX problem-tutorial lessons are an example: students were frequently seen using the hard copy "notes" generated during the lessons to prepare written solutions to the problems.

The student terminals were first placed in an area adjacent to the Chemistry laboratories, but were later moved to a location in the University Library where they would be accessible to more students. The terminals were available seven days and five evenings a week, and an attendant was always available to help students sign on and use the system. We found that the presence of a personable and sympathetic attendant was quite important to allay the anxiety that many students feel when they first encounter a computer terminal. Another essential function of the attendant was to identify program bugs or errors that became apparent during student use of the programs.

The Coursewriter authoring language proved to be the primary constraining influence on the quality and kind of CAI we were able to deliver. Although it does offer many important conveniences, it also has many limitations that were tolerable at first, but became less so as our experience with CAI developed and our requirements became more sophisticated.

In late 1972 we decided that Coursewriter could no longer meet our needs; since our Computing Centre had just implemented York/APL, we decided to use this language for CAI instead. APL, in spite of its power and flexibility, is definitely not an instructional programming language; we therefore attempted to retain some of the advantages of Coursewriter by simulating some of its features (such as implicit branching) in APL. This required not only the writing of special APL functions, but also some basic changes to the APL system itself, including the provision for accessing assembly-language subroutines to perform certain operations that are extremely inefficient in APL.

The resulting APL-based authoring language was far more satisfactory, and enabled us to write better programs with much less difficulty than with Coursewriter. The main gain was the ease of carrying out calculations and specifying random selection of problem parameters.

One disadvantage of going over to such a system was that we were no longer able to exchange programs with other institutions. In practice this did not affect us seriously, since we have never found other centers to be an important source of useable programs.

Although our APL-based system represented a definite advance over Coursewriter from the standpoint of the CAI author, a number of systems difficulties began to manifest themselves. York/APL is relatively inexpensive and requires only a small amount of core, but it is not very efficient, it is incompatible with other implementations of APL, and an undue amount of local maintainence was required to keep it going. At about this time, IBM's new APLSV became available, and it was decided to drop York/APL in favor of this implementation.

APLSV offers most of the features that York/APL had and that APL/360 lacked (a notable exception, crucial for CAI, is error trapping). Although we could not access assemply-language functions through APLSV, we hoped its much greater efficiency would obviate the necessity of doing this. After some experimentation, however, we found that some of the Coursewriter-like operations we wanted were still intolerably slow even in APLSV, and that the large number of utility functions required would leave inadequate room for instructional material in the student workspace.

As a result of these and other considerations, we decided that APLSV cannot serve as a practical vehicle for the authoring and delivery of CAI at the level of quality and sophistication that we feel is required to justify the time and effort that must be devoted to CAI. Since the only other CAI authoring language

that would be adequate is TUTOR, and this language is suited only to the PLATO system, we decided that the only practical solution would be to develop our own programming language. Some comments on this language are given in the last section of this article.

While we were experiencing reasonable success in the use of CAI in the teaching of my Chemistry course, there is another side of the picture that was somewhat less encouraging. This was that after several years, and even after the beneficial aspects of CAI in the introductory course had been demonstrated, I was still the only faculty member in my Department and in the University who was regularly using CAI in regular courses. Although students and some faculty had written CAI modules in several other fields (English, Education, Commerce, Mathematics), these tended to be used either very irregularly or not at all, and in most cases were not developed to the point that they could serve as an integral part of an ongoing course. As a result of this "lack of faculty interest", the University decided to discontinue support of this facility, and CAI service was dropped in 1975.

WHAT WE HAVE LEARNED

Since our work in CAI started out as a pragmatic attempt to apply CAI to a real teaching environment rather than as a "research project", we have had neither the time nor the inclination to gather the quantitative data that would provide a complete picture of the use of CAI and its results. All I can offer are a number of admittedly qualitative and subjective impressions that have developed during our six years of CAI experience.

Most of these are conveyed in Parts I and III of this article, so only a few additional comments need be made here. First, and most important, I discovered that writing CAI lessons and making them work is a full time job in itself; even after the programs are written and put into use, there will always be minor bugs to fix and modifications to be made on the basis of student experience and performance.

It is always wise to design the CAI programs so that they can be used and combined in many different ways, rather than writing them in a way that is uniquely adapted to only one particular course. This usually means a modular approach, although the individual modules may not necessarily be apparent to the students.

The students' view of CAI was interesting; not unexpectedly, they were far more enthusiastic about it than were most of the faculty, and generally asked for "more CAI" when offering suggestions for improving the course. We feared at first that many students would view CAI as a too "impersonal" form of instruction. We soon found, however, that this was a very rare complaint; comments we frequently received were that "CAI lets you make your mistakes in private", and that "being in a lecture room with one hundred other students is no less impersonal". To some extent, the fact that CAI reduced the number of students seeking personal help to the relatively small fraction of the class who experienced serious difficulties, made me and the other course staff more available than we otherwise would have been.

CURRENT DEVELOPMENT

It would indeed be unfortunate if, after having accumulated six years of experience in using CAI materials in two different authoring languages, we were to abandon the work at this stage--particularly when we finally have available a powerful and convenient authoring language. Consequently, for the past year I have been constructing a collection of CAI programs known as GENCHEM, designed to support an entire course in General Chemistry at the First-year college level; much of it can also be utilized in high school courses. Rather than being a pre-packaged "course" in Chemistry, GENCHEM instead provides the basic materials and data from which the instructor can construct a learning aid that is tailored to the particular needs of a given course.

GENCHEM is written in a highly modular manner, so that the lessons can be of maximum use in a variety of settings. The materials are classified into about 25 different subject groups, each of which contains modules that variously emphasize tutorial or drill-and-practice interaction at several levels. The various components can be accessed in different ways: as either self-tests, drills, or "mini-courses" on topics selected by the student, or additionally as pre-defined segments keyed to the schedule and organization of a particular instructor's course.

To aid teachers who wish to use GENCHEM, extensive documentation is being designed that will clearly state what each component of GENCHEM does, and what pre-knowledge it assumes the student will have.

In addition to its role as a set of operating courseware, GENCHEM is intended to serve as a source collection of computer-based instructional strategies that can be used by others who wish to design CAI materials of their own, perhaps written in another programming language. The programs are written in a high-level language that should be fairly easily comprehended by anyone who has a good understanding of program flow in CAI. The program listings themselves should therefore constitute another level of documentation that will convey the general sense of the programs, so that a competent programmer should be able to implement the programs in another authoring language, particularly if a reasonably powerful text editing program is available to minimize the amount of additional keyboard entry required. Naturally, any such "translation" will be within the limits imposed by the local language. This may be a serious practical limitation to many, particularly if the local language is not particularly sophisticated. Use of the very simple CAI languages or non-CAI languages such as APL or BASIC will probably not be practical unless a very large amount of programming manpower is available. Even then, the results are not likely to be satisfactory, since the design of the GENCHEM strategies presupposes the availability of facilities that are not ordinarily available on general-purpose interactive systems.

Similarly, a certain standard of terminal capability is assumed. Ability to present both upper and lower-case letters is essential. About two-thirds of the programs can be used on either printing or display terminals; the remainder are adapted specifically to display terminals with individual character addressing.

It is hoped that the GENCHEM materials, when they are completed, will help to alleviate the severe shortage of adequate course material that must certainly militate against the more widespread use of CAI in the teaching of Chemistry. Of course there will be many who will still be unable to make use of this material owing to the lack of sufficiently sophisticated systems and terminals. Given the rapid development of computer technology, it does not seem wise to sacrifice quality and sophistication of courseware in order to meet this short-term difficulty.

It is hoped that GENCHEM [6], even if not immediately implementable by others, will at least prove to be a useful model for constructing comprehensive CAI packages in other disciplines, and a basis on which others can improve upon the art of CAI in Chemistry.

Part III. TUTORIAL CAI: AUTHORING LANGUAGES, AUTHORS, AND TRANSFER

If the purpose of computer-assisted instruction is to communicate with the individual student, then the function of a CAI authoring language must be to communicate both the instructional material and the instructional strategy to the computer. At first glance, one might expect that a general-purpose computational language would be the ideal vehicle for expressing an instructional program, since most such languages can manipulate and analyze character strings and perform virtually any kind of calculation that might be required. In fact, the use of such languages for anything but the simplest kinds of CAI programs is extremely cumbersome and usually leads to unsatisfactory results. In most cases, the very generality of the language becomes a severe impediment; the algorithms required to implement many of the simplest instructional operations are either hopelessly complex, or else they require the insertion of large amounts of repetitive code. Part of the problem arises from the strict syntax rules that must necessarily be tightly drawn in any language that is to serve a multiplicity of uses, but which simply get in the way of convenient CAI practice.

These deficiencies are particularly serious in a relatively primitive conversational language like BASIC, in which the number of fundamental operations is quite limited, and variable types cannot be dynamically defined. But even in the far more elegant and powerful APL, many of the very features that give this language its simplicity and integrity lead to constructions and restrictions that tend to distort a direct and natural instructional programming style. APL can be used for CAI in its more or less "raw" form, or as a higher-level language in varying degrees of disguise. The resulting programs are generally difficult to debug or modify, are not easily transportable between systems of different manufacture, and the listings of the CAI programs are very difficult to follow. The fact that APL has been used for CAI with some success is more of a tribute to the ingenuity and perseverence of the authors and programmers, than an indication of APL's suitability for this application [7].

General purpose languages are also awkward for students to use in an instructional context; it is rarely practical to "trap" simple errors such as input of an alphabetic quantity when a numeric answer is required, nor is it usually possible to allow a student to enter expressions for trial evaluation, or to intercept special student commands (such as "help") without making special provision at each input point of the program.

There are also problems arising from the limited size of the active workspace or memory region, requiring either frequent "load" and "run" commands on the part of the student, or that the author go to special lengths to manage the use of active memory.

The obvious alternative is to use a specialized language in which simple statements or commands are available to invoke a wide spectrum of operations peculiar to CAI. These include the organization of material into "question groups", analyzing complex student responses, implicit branching, and responding to student-entered commands.

Many such languages have been developed since the early- and mid 1960's. Most of them were developed for use on a single computer system for local use. Some languages were designed for more general use, and a small number of these have received fairly widespread acceptance. Perhaps the most widely used of these languages is IBM's Coursewriter III. Coursewriter is usually considered easy to learn and to use for persons who are not computer-oriented, but its lack of flexibility and extensibility limit its use to rather simple strategies that require little computation. In spite of Coursewriter's many advantages, it has failed to evolve along with computing technology and CAI practice, and its future application seems limited to fairly simple training applications.

The most powerful of the current CAI languages is TUTOR, which was developed as part of the PLATO project at the University of Illinois [8]. In contrast to Coursewriter, the development and use of TUTOR have been tightly coupled, so that the language has been able to evolve of the users. Although TUTOR does have its awkward aspects, it is nevertheless one of the most sophisticated CAI languages, and the only one that supports interactive graphics. Because TUTOR reflects the unique architecture of the PLATO system, its use is presently limited to that system, which is currently being offered by the Control Data Corporation.

PLANIT [9] is a language that was developed with the support of the U.S. National Science Foundation. This language is roughly comparable to Coursewriter, except that it has full numerical computational facilities and allows the use of subroutines. In the interests of transferability, PLANIT was written in FORTRAN. Nevertheless, PLANIT has not received wide acceptance, and it now seems unlikely that it will remain a language of major importance.

A fourth language that bears mentioning is PILOT. Although PILOT was developed in an attempt to create a "simple" authoring language, its principal application has been by users of minicomputers, and its implementation is usually associated with the availability of BASIC. The principal drawback of PILOT is that its capabilities are generally limited to those of the smaller minicomputer systems, which are only marginally suitable for many CAI applications. As computer technology evolves, there should be relatively little need for primitive languages such as PILOT (or even BASIC, for that matter!) for serious CAI work.

"SIMPLICITY" VS. ORGANIZATION

Virtually all CAI authoring languages seem to be described (by their originators, at least!) as "simple" and "easy to learn". Aside from the questionable objective meaning of these terms, it seems reasonable to ask if they are really valuable features at all. Certainly, no one would argue that a CAI language should not be simple to use if the programming task itself is simple from a computational standpoint, as in a fairly linear "question-and-answer" kind of CAI program. But it can be argued that CAI programs of this sort represent a small fraction of what most of us will soon want to do with CAI, and they will certainly fall far short of exploiting the full power (and cost-effectiveness) of the medium, and above all the capabilities of the students who will be using the medium.

A language can be exceedingly simple in terms of number of commands, and thus "easy to learn"; at the same time, it might be almost as difficult to use for reasonably sophisticated work as a non-CAI language might be. The point that must be made is that the instructional process itself is far from "simple", particulary when stated in terms of basic computer operations. A better index of the adequacy of a CAI language would be the amount of <u>organization</u> it offers to help deal with this complexity. It is in precisely this area that the "simple" languages like PILOT, and the non-CAI languages like BASIC and APL clearly fall down. It is not so much a question of WHAT the "simple" languages can do, but how much and what kind of code must be written to get them to accomplish a given instructional task.

It seems inevitable that more and more use will be made of minicomputers in CAI; not only are they cheaper to operate and more reliable, but they are not plagued by "computer politics" and bureaucratic administration to the extent that large central

computing systems are. Although few if any of the present-day mini systems may be capable of supporting a really sophisticated CAI authoring language, a fairly significant fraction of them may well be capable of executing the "compiled" code that might be generated by an adequate authoring language on a much larger machine. In other words, since there will inevitably be more users than authors, and since it is the authoring process that demands the larger machine, it would seem worthwhile to put more emphasis on the separation of these two aspects of CAI. One can envisage systems in which one large computer serves an extended network of minicomputers, each of which handles a number of local terminals. Each mini would communicate with the central machine to receive updated material, and also to transmit performance and use data relating to the programs.

TEACHERS, AUTHORS, AND PROGRAM DOCUMENTATION

It is not unlikely that almost every teacher will one day have access to a personal computer terminal. Others have noted this, and have suggested that this would enable every such teacher to write CAI programs specific to the needs of each class. To me, such a possibility seems no more likely (or desirable) than to expect everyone owning a typewriter to write a textbook. The reasons are the same: the successful CAI lesson, like the successful textbook, requires an author who possesses a good amount of experience and talent, both in designing and using the particular medium, and also in teaching. Both endeavors, and CAI especially, also consume massive amounts of the author's time.

I firmly believe that CAI is (or should be) the art of the teacher, not of the computer programmer. Only the best teachers should be encouraged to develop CAI material. Unfortunately, the number of sufficiently good teachers who are also able and willing to devote the required amount of time and effort to this work is vanishingly small. The temptation to short-circuit the authoring process by leaving most of the work to student assistants or professional "instructional programmers" is always present, particularly when one must rely on an inadequate programming language whose use is accompanied by a good deal of drudgery. The results of succumbing to this temptation to any significant extent are usually apparent in the quality of the finished product. To compose a textbook or a lecture in this manner would surely result in an inferior product. One can only wonder if the rather poor image that CAI has in many quarters does not arise in part from this tendency to regard CAI as a "programming" discipline.

It seems likely that the bulk of the successful CAI materials will be written by a fairly small group of teacher-authors. Because the structure and organization of all but the most "Mickey Mouse"-ish of instructional programs is fairly complicated, these authors will of necessity possess considerable programming expertise as well. Those who want to use CAI in a major way will rely largely on the "import" of these materials from other sources. Naturally, we can always modify, extend, or otherwise supplement these materials to adapt them to our own needs and idiosynchrosies, just as we frequently supplement a textbook with locally-printed written material.

It would therefore appear that the widespread acceptance and use of CAI in a meaningful instructional sense will not be possible until tested and demonstrably "effective" CAI materials become generally available; indeed, it is just the lack of this availability that was cited as a principal inhibitory factor to the spread of CAI, in a study conducted several years ago [10]. More important, the very development of the art of designing CAI materials depends on new and prospective authors having access to existing programs, for use as models; certainly very few conventional authors would commence the design of a new textbook without surveying the best (and worst) of what has been done in the past.

A book, of course, serves as its own documentation; by simply reading it, one can immediately grasp the author's style of approach, the scope of coverage, and its level of sophistication. The same cannot be said of most CAI programs. Any but the trivially simple ones cannot conveniently be "read" like a book, even if the person looking at the printed listing is familiar with the authoring language. In any case, the teacher who is interested in assessing the program is really interested not in the program listing itself, but in the end product of the program, which has no single representation, since any such program will normally specify a multiplicity of actions that will be contingent on the student responses, and perhaps on randomly- specified parameters within the program.

CAI program portability in the broadest sense will require that a number of conditions be satisfied. First, the program listing itself should be as clear and self-documenting as possible. This implies a high-level authoring language whose power and organization is fully consistent with the sophistication of the CAI program. There should be an absolute minimum of detailed computational code, and this should be adequately documented internally. Similarly, the overall organization of the program, meanings of variables, and default conditions should be clearly noted within the program.

Second, there will usually need to be a considerable amount of "external" documentation that describes what each program attempts to do, what prerequisites are assumed, what questions are asked, and what kinds of errors are looked for. Where problems are constructed from randomly-defined parameters within the program, examples should be given that indicate the general range of the different variables. Similarly, the contents of various tables of data that are used in the program should be given in clear, understandable form, so that one can see what the scope of the program will be.

A third level of documentation would cover the actual use of these programs in connection with the overall course. It would suggest ways of correlating the CAI lessons with reading, audio-tutorial exercises, and other class work. If alternative ways of organizing the program are available, these would be explained. This third level of documentation would be a "User's Manual" for the teacher, and would be an important means of combatting the natural suspicion that most instructors harbor toward instructional materials that they have not designed themselves, and which they perhaps do not fully understand.

It is apparent that all this will require a large amount of additional work by the author, not only to prepare, but to revise and update as the CAI programs themselves evolve. It is doubtful, however, that the fruits of any one author's work will ever diffuse to a very wide audience if this is not done. And it is hardly likely that many authors will be inclined to devote several years of nearly full-time effort to the creation of a major group of programs unless there is a reasonable chance that their work can be used by others, whether the return to the author will be purely professional, or financial as well [11].

AUTHORING LANGUAGES AND PROGRAM PORTABILITY

What we have been discussing so far is the transfer of the description, or idea of the program. What of the program itself? The most direct and trouble-free route is of course to transfer between identical computer systems, although it must be pointed out that very few systems are totally identical in all respects. The opportunity for divergence increases rapidly with system size and complexity; frequently the differences between such things as terminal telecommunications control protocols is enough to raise serious difficulties in the transfer of interactive programs.

Many of the largest installations that may be most interested in sharing materials will very likely be committed to their own, and frequently very different authoring languages. This raises a serious impediment to transportability, and there are no simple solutions, short of the utopian one of hoping everyone will adopt the same language.

It has been suggested that authors might make their major programs available in a "documentation language", which would then be translated by interested users into whatever local language is available. A model language has been proposed for this purpose [12].

The major difficulty that arises when languages or systems differ is that the capabilities that the original CAI program requires may simply not be available at the receiving location. For widespread transportability and use, programs must be written in a widely available authoring language that will necessarily reflect a common standard of program sophistication and capability. With all commonly-used authoring languages (except for TUTOR), this standard is entirely too low; it does not reflect the functional capabilities of present hardware or software practice, and it certainly inhibits the realization of CAI's potential. The best programs, the ones that involve years of development work on the part of the author, may well require features that are unavailable in the majority of standard authoring languages, and for this reason they are not easily transferable. The less imaginative, more trivial, (and more "machine-like") programs are the ones that are most subject to exchange. Thus with the present low standards in programming languages, the transfer process does little to advance the state of the art; instead, it promotes obsolescence and mediocrity.

Of course, creative teachers will always be getting better ideas and computers will continue to evolve, so the completely unimpeded tranfer of programs is not a realistic goal. What would seem necessary at this stage is the setting of a minimum standard of functional capability that is reasonably consistent with the best of present CAI practice. This has already been done in Canada, under the sponsorship of the National Research Council [13], and a CAI authoring language consistent with this standard has been designed [14]. It is still too early to know whether this work will ever be translated into practice, but these are certainly moves in the right direction.

A NEW C.A.I. AUTHORING LANGUAGE

Rather than enter into a discussion of what features I believe the "ideal" CAI authoring language would have, it might be more useful to refer interested readers to one practical approach to that ideal. We have been developing a new authoring language [15] which we feel is far more consistent with present needs and practice than are most of the other languages currently available. It evolved from our own direct experience in using Coursewriter III and an APL-based CAI language [16]; it also draws heavily on TUTOR, and also incorporates many features fouund in such non-CAI languages as APL, SNOBOL4, and PL/I. Above all, it reflects the needs that we have perceived in six years of actual use of CAI in real teaching environments.

This language offers several novel features, the most notable of which are briefly listed below:

1. The language is very nearly self-documenting; the listings are quite easy to follow by anyone who has a general understanding of program flow in CAI. Documentation is materially aided by the module structure, the use of DEFINE statements for local and global variables, and the ease of inserting comment lines or comments within lines. These characteristics should significantly advance the cause of courseware portability and generality.

2. While the language is designed to operate interpretively, the programs are first compiled. This yields a considerable increase in operating efficiency (and also generates a good deal of useful debugging diagnostics) without any serious inconvenience to the author/user.

3. It is the only CAI language that offers the major features of Coursewriter, most of the primitive operations of APL (without the special symbols), and a number of TUTOR-like operations, including the very powerful conditional verb specification and the JOIN statement. As in TUTOR, the language is procedure-oriented; this increases the flexibility of the language and enhances the organization and readability of the program.

4. In contrast to the traditional practice of writing a separate system for terminal telecommunications control and for editing of the CAI lessons, our language is designed to take full advantage of the general purpose systems

that are now being more commonly used for these purposes. Thus terminal I/O can be via CMS or MILTEN. Authoring and editing can be done on any on-line text editing system such as WYLBUR [17].

5. Many applications of CAI require access to external data files, such as tables of atomic weights, lists of sentences, etc. The construction and access of such files is easily accomplished in the present language, without recourse to complicated coding on the part of the author. In general, the author need not be concerned with details of memory management, data access or storage.

6. The language meets the major requirements of the functional standard for CAI languages that has been adopted by the National Research Council of Canada. The language is also easily extensible in the sense that the NRC Standard recommends.

- Compared to other CAI systems of comparable power, the new language makes surprisingly modest demands on computer resources: although our implementation on the 370/155 system is still not complete, it is estimated that the system will occupy a core partition of less than 40K, exclusive of terminal telecommunications control.

Whether or not this language (which is as yet unnamed) ever gains wide acceptance is perhaps less important than the fact that it provides a vehicle for writing CAI materials at the present time. This is, after all, the rate-limiting step in the evolution of CAI, and in my view it should not be deferred until an "ideal" language comes into wide use. Even if the resulting materials cannot enjoy wide application initially, an author can at least invest his time in a kind of CAI that represents the present state of the art, rather than that of a decade ago. And eventually, when a suitable language does come into wide use, it will certainly have most of the features of the present one, so that translation should be a fairly trivial problem, given the power of modern text editing systems.

Perhaps the most important contribution that this work can make to the CAI community at large will be to demonstrate that significant improvements in present practice are not only possible, but are also quite practical on computer systems of reasonable size.

REFERENCES

1. N. Flitcroft, W. Wong, and B.D. Pate: "Audio-visual aids in a freshman laboratory course". Chemistry in Canada, September, 1967.

2. Samuel N. Postlethwait, et. al. "The Audio-tutorial approach to Learning", 2nd. Edition, Burgess Publishing Co., 1969.

3. S.K. Lower, "Audio-tutorial and CAI aids for problem solving in introductory Chemistry". J. Chem. Education 47 (143) 1970.

4. S.K. Lower: CHEMEX and ALCHEM Instructor's Guides. Simon Fraser University, 1973. The SFU Coursewriter III programs are freely available through the clearinghouse operated by SECOS, 420 Main St., Poughkeepsie NY 12601.

5. S.K. Lower, "Chemical Calculations: An Audiotutorial Approach". Simon Fraser University, 1974. (Instructor's guide, Problem Book and sample cassette available from the author).

6. S.K. Lower: "Guide to Genchem: Computer-assisted instruction for general chemistry". Preliminary draft edition, approx. 80 pages. Simon Fraser University.

7. S.R. Peterson: "Unsuitability of APL for CAI". Simon Fraser University Computing Centre Report No. 34 (1975).

8. The literature on PLATO is extensive. See, for example, S.G. Smith and B.A. Sherwood: "Educational uses of the PLATO computer system" Science, 192 (p 344-352) 1976.

   For a description of the TUTOR language, see B.A. Sherwood: "The TUTOR Language". Computer-based Education Research Lab, University of Illinois, Urbana, Ill.

9. C. Frye: "Report on PLANIT: One Stage of Completion". ERIC Document Reproduction Service, P.O. Box 190, Arlington VA 22210

10. E. Anastasio and J. Morgan: "Factors inhibiting the use of computers in Instruction". EDUCOM, Princeton, N.J., 1972.

11. For a lucid discussion of authors' rights that seems to have direct application to CAI, see C.N. Mooers, "Computer Software and Copyright", Computing Surveys 7 (p 44), 1975.

12. P. Ripota: "A Concept for a Primary Author's Language" (3 volumes); Projekt CUU, 78 Freiburg, Schaenzlestr. 9-11, West Germany

See also, J. Schuyler, "Computer Augmentation of the CAI Courseware Authoring Process-- the CAI Design System", to be published in the Journal of Computer Based Instruction.

13. "A functional specification for a programming language for computer-aided learning applications". National Research Council, Ottawa, Canada 1972.

14. NATAL-74 Author's Manual and Reference Manuals. National Research Council, Ottawa Canada, 1975.

15. S.K. Lower and S.R. Peterson: "A Guide to the S.F.U. CAI Authoring Language". Draft version. Simon Fraser University, 1976. (Approx. 65 pages)

16. S.K. Lower and S.R. Peterson: "A CAI Language Based on APL". Simon Fraser University, 1974. This document describes a language having a Coursewriter-like structure that was implemented in APL.

17. R. Fajman and J. Borgelt: "WYLBUR: An Interactive text editing and remote job entry system." Comm. ACM 16 (314) 1973.